

```

class Regressor0(object):
    def __init__(self, arrayTheta, alpha, x, y, maxIt, epsilon, reg):
        # Inicialització segons els paràmetres
        ...
        #atributs per analitzar comportament dels paràmetres
        self.costos=[]
        self.iterations=[]

    def predict(self):
        loss = np.dot(self.x, self.arrayTheta)
        cost = np.sum(loss ** 2) / (2 * self.x.shape[0])
        return loss

    def __update(self, hy):
        self.arrayTheta = self.arrayTheta*(1-self.alpha*
        (self.epsilon/self.x.shape[0])) -(1/self.y.shape[0])*alpha*( self.x.T.dot((hy -
        self.y)))

    def train(self):
        i=0
        millora=epsilon+1
        costAnt=0
        while( (i <= self.maxIt) and (millora > self.epsilon)):
            self.it+=1
            pred = self.predict()
            loss=pred-self.y
            self.cost=(np.sum(loss**2)+reg*np.sum(thetas**2))/(2*self.x.shape[0])
            self.costos.append(float(self.cost)*100)
            self.iterations.append(self.it)
            millora = abs(self.cost-costAnt)
            self.__update(pred)
            costAnt=self.cost
            print("Iter: "+str(self.it)+" Cost: "+str(self.cost))
            i+=1
        return self.arrayTheta

```